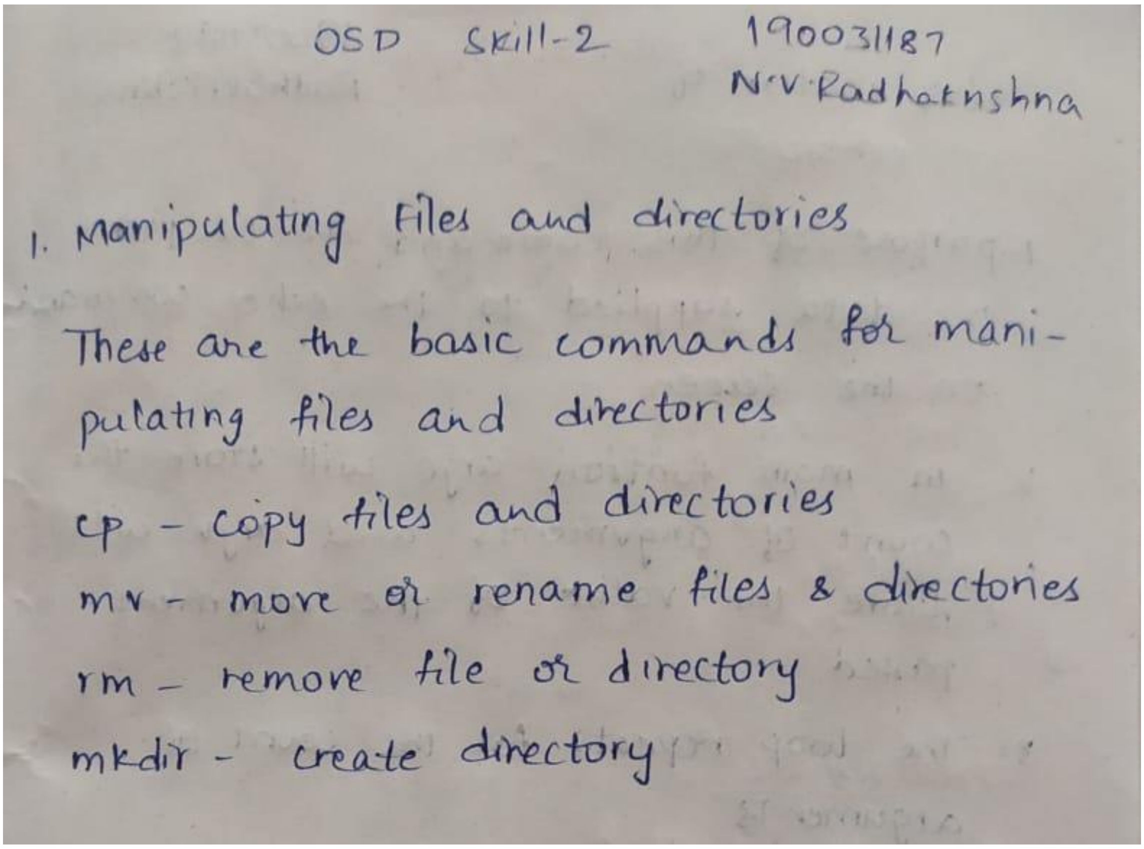
Operating System Design – 19CS2106S

Skill Experiment - 2

1) Manipulating Files and Directories.



OSD SKill-2 190031187
N·V·Radhakrishna

1. Manipulating Files and directories

These are the basic commands for manipulating files and directories

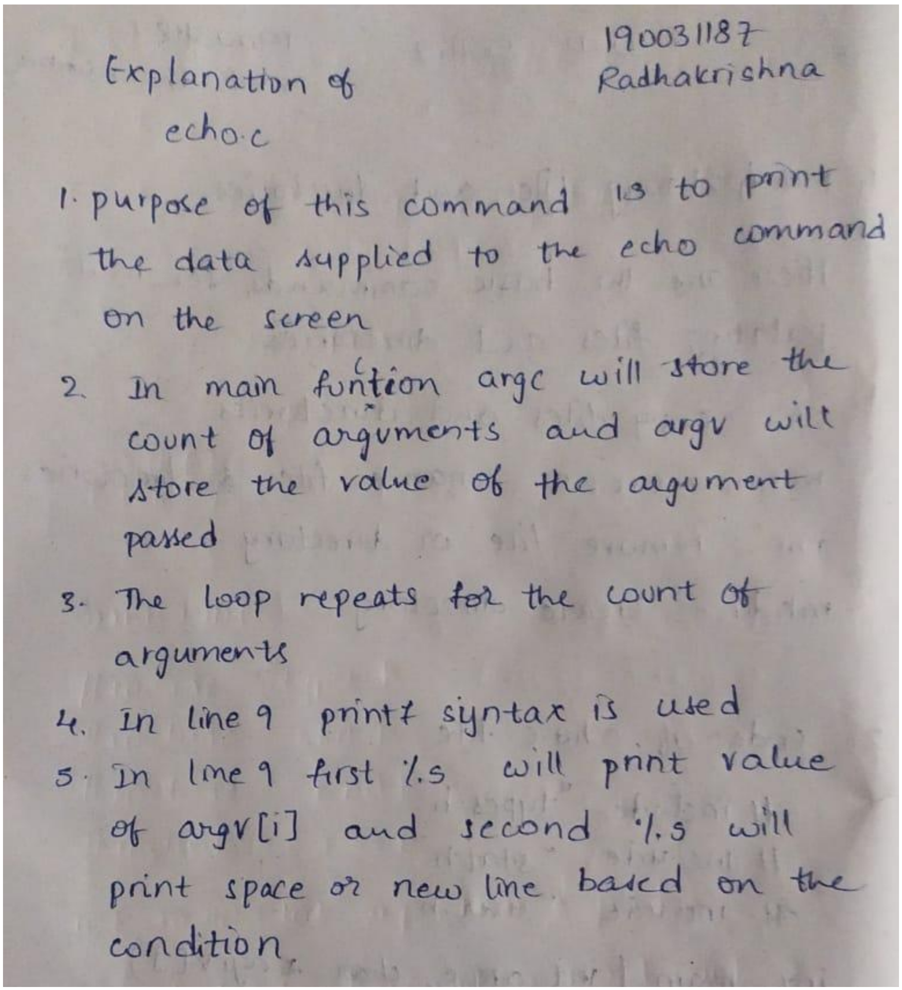
cp - copy files and directories

mv - move or rename files & directories

rm - remove file or directory

mkdir - create directory

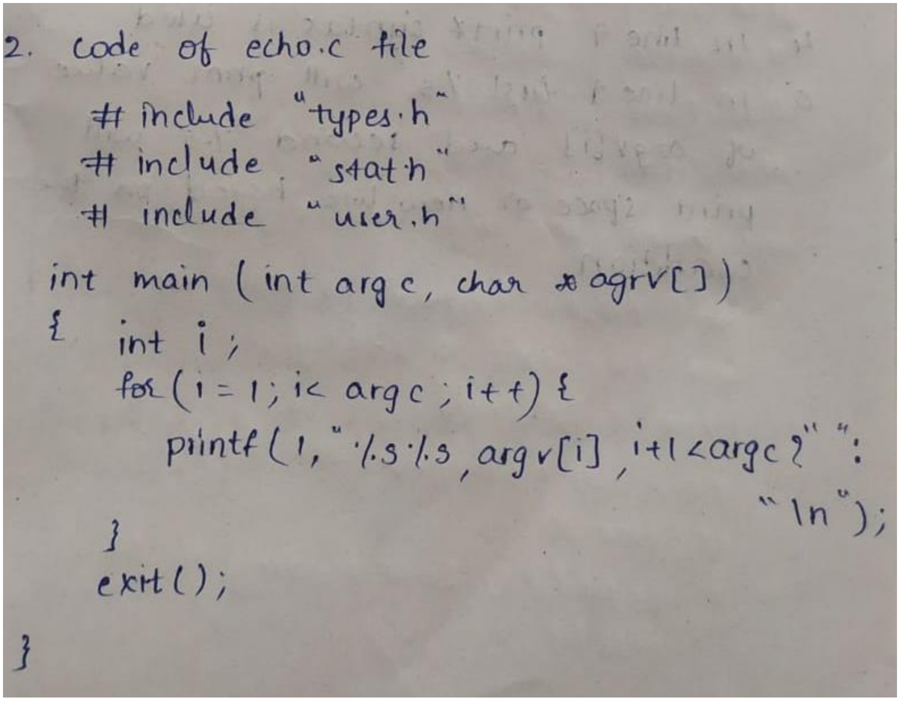
2) echo.c(xv6 design implementation)



Explanation of
echo.c

190031187
Radhakrishna

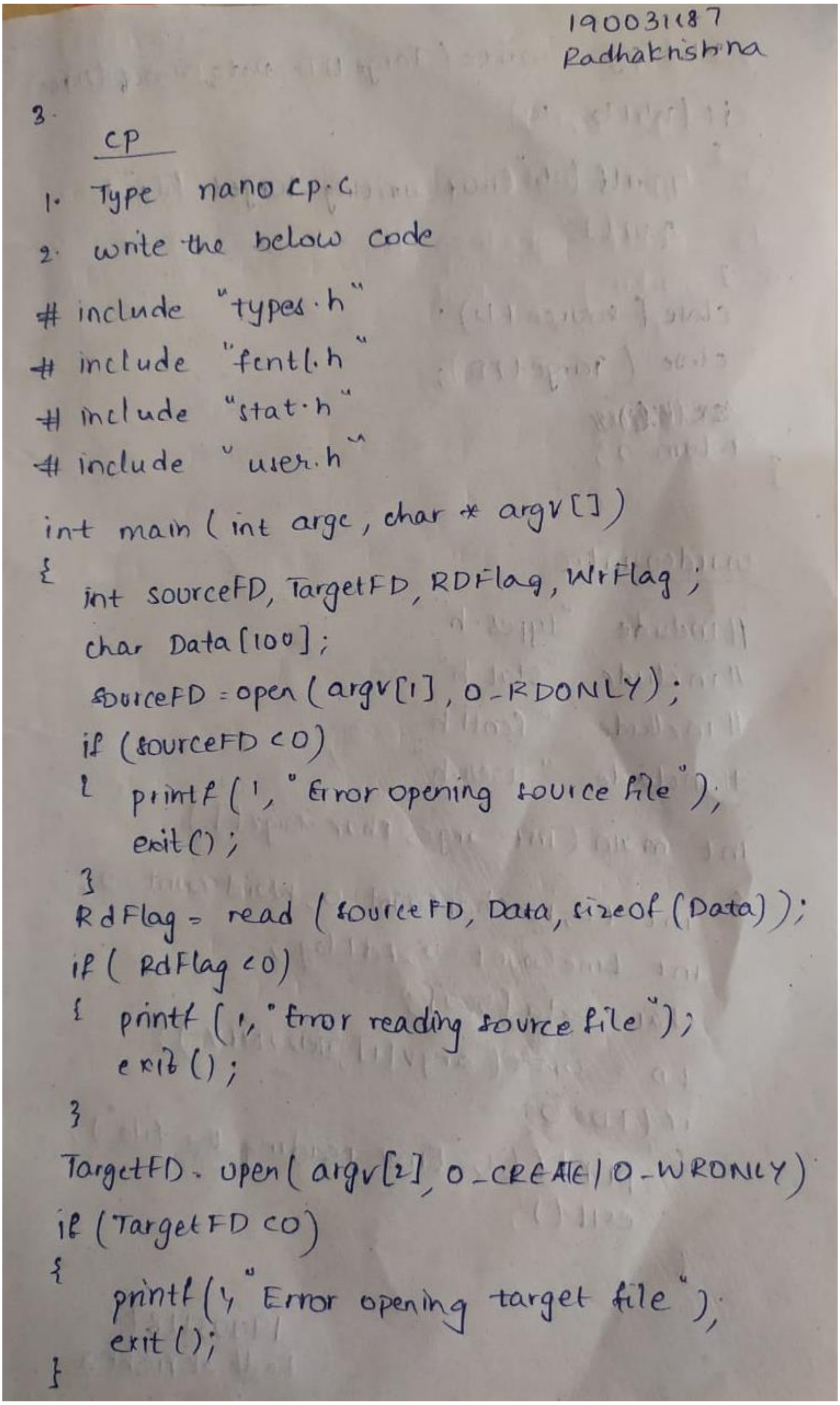
1. purpose of this command is to print the data supplied to the echo command on the screen
2. In main function argc will store the count of arguments and argv will store the value of the argument passed
3. The loop repeats for the count of arguments
4. In line 9 printf syntax is used
5. In line 9 first %s will print value of argv[i] and second %s will print space or new line based on the condition.



2. code of echo.c file

```
  # include "types.h"
  # include "stat.h"
  # include "user.h"
int main (int argc, char *agrv[])
{  int i;
   for (i=1; i<argc; i++){
      printf(1, "%s%s, argv[i], i+1<argc?" ":
                                        "\n");
   }
   exit();
}
```

3) cp, wc.c (xv6 customization)



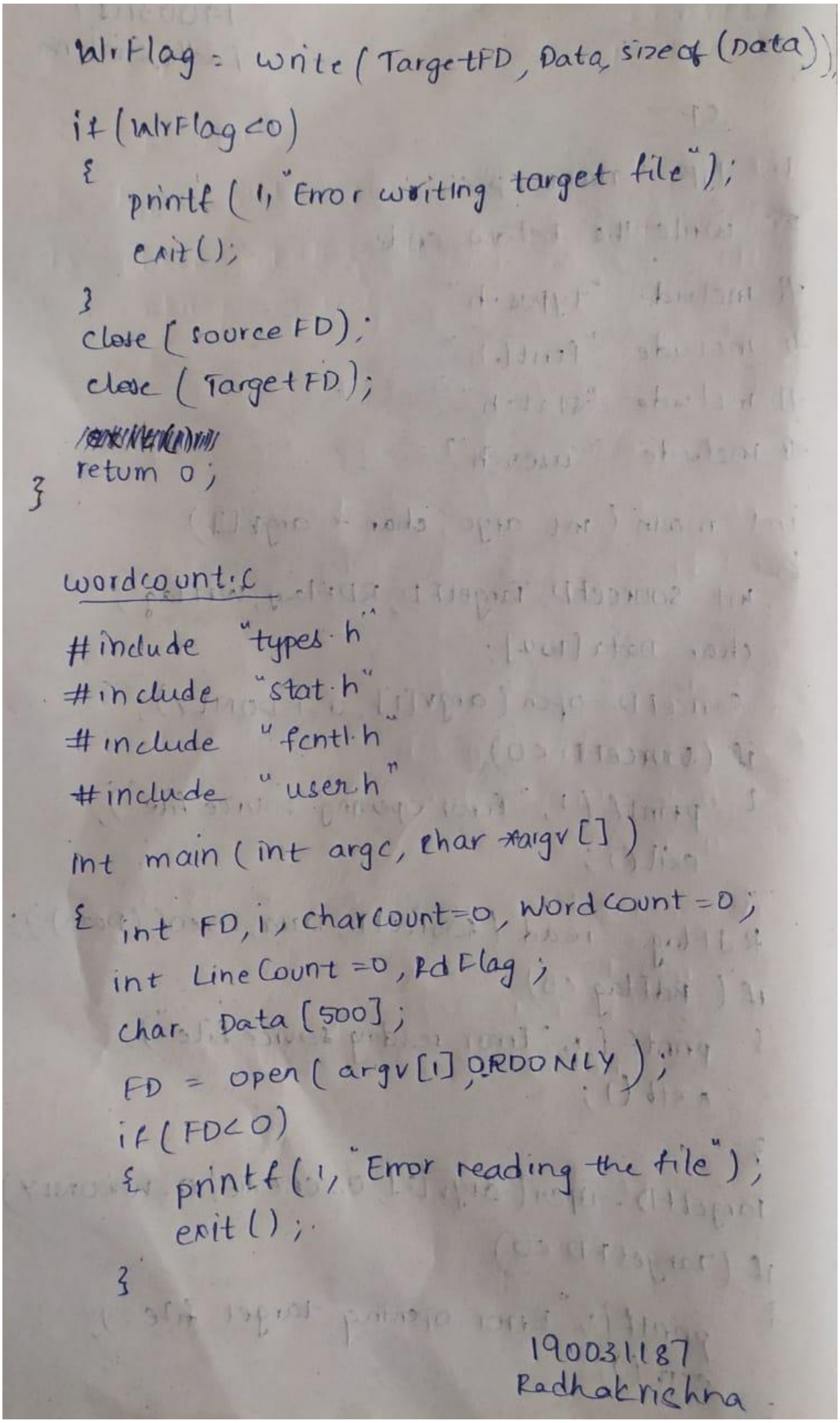
190031187
Radhakrishna

3.

cp

1. Type nano cp.c
2. write the below code

```
# include "types.h"
# include "fcntl.h"
# include "stat.h"
# include "user.h"
int main ( int argc , char * argv[])
{
   int sourceFD, TargetFD, RDFlag, WrFlag ;
   char Data[100];
   sourceFD = open (argv[1], O_RDONLY);
   if (sourceFD <0)
   {  printf (1, "Error opening souce file");
      exit();
   }
   RdFlag = read (sourceFD, Data, sizeof (Data));
   if ( RdFlag <0)
   {  printf (1, "Error reading source file");
      exit ();
   }
  TargetFD = open( argv[2], O_CREATE|O_WRONLY)
  if (TargetFD <0)
  {
     printf(1, "Error opening target file");
     exit();
  }
```

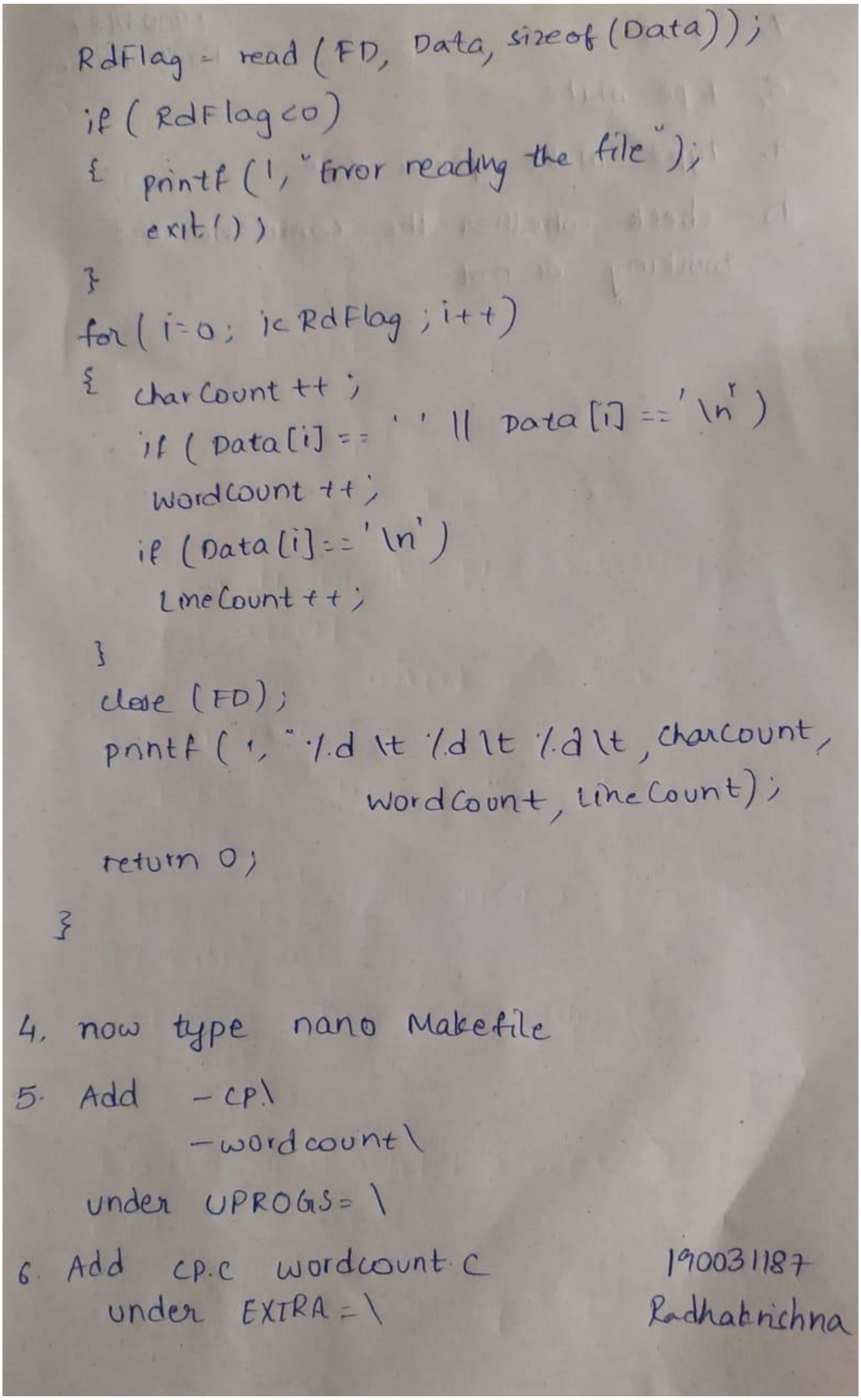


```
WrFlag = write(TargetFD, Data, sizeof(Data));
if(WrFlag<0)
{
   printf(1,"Error writing target file");
   exit();
}
close(sourceFD);
close(TargetFD);
return 0;
}
```

wordcount.c

```
#include "types.h"
#include "stat.h"
#include "fcntl.h"
#include "user.h"
int main(int argc, char *argv[])
{ int FD,i, charCount=0, WordCount=0;
  int LineCount=0, RdFlag;
  char Data[500];
  FD = open(argv[1],O_RDONLY);
  if(FD<0)
  { printf(1,"Error reading the file");
    exit();
  }
```

190031187
Radhakrishna



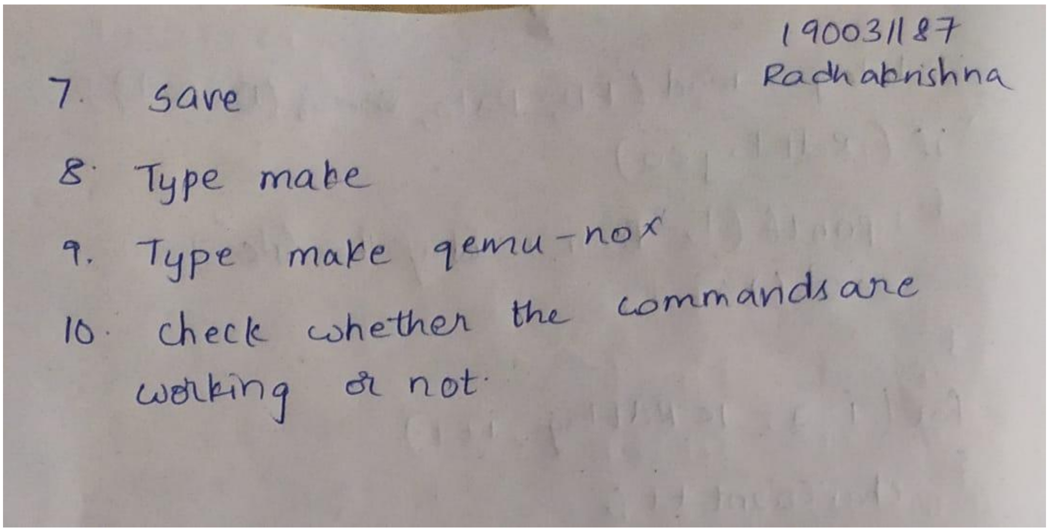
```
RdFlag = read (FD, Data, sizeof (Data));
if ( RdFlag<0)
{  printf (1, "Error reading the file");
   exit())
}
for( i=0; i<RdFlag ; i++)
{  charCount ++ ;
   if ( Data[i] ==  ' ' || Data[i] ==' \n' )
   wordCount ++;
   if (Data[i]==' \n')
   LineCount ++;
}
close (FD);
printf ( 1, " %d \t %d\t %d\t, charCount,
              wordCount, LineCount);

return 0;

}
```

4. now type nano Makefile

5. Add -CP\
   -wordcount\
   under UPROGS= \

6. Add CP.C wordcount.C
   under EXTRA =\

190031187
Radhakrishna



190031187
Radhakrishna

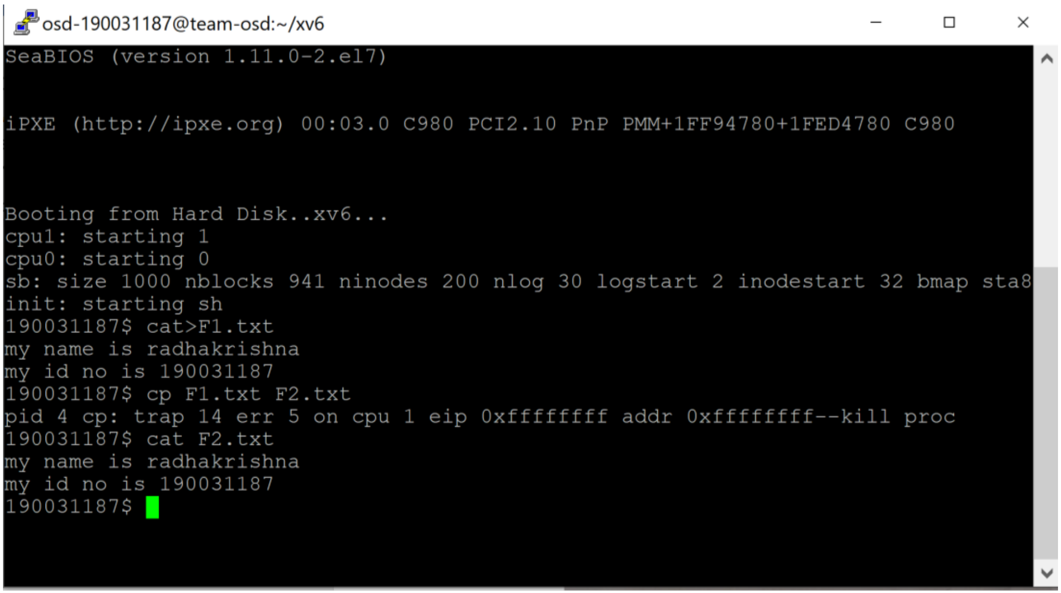
7. save

8. Type make

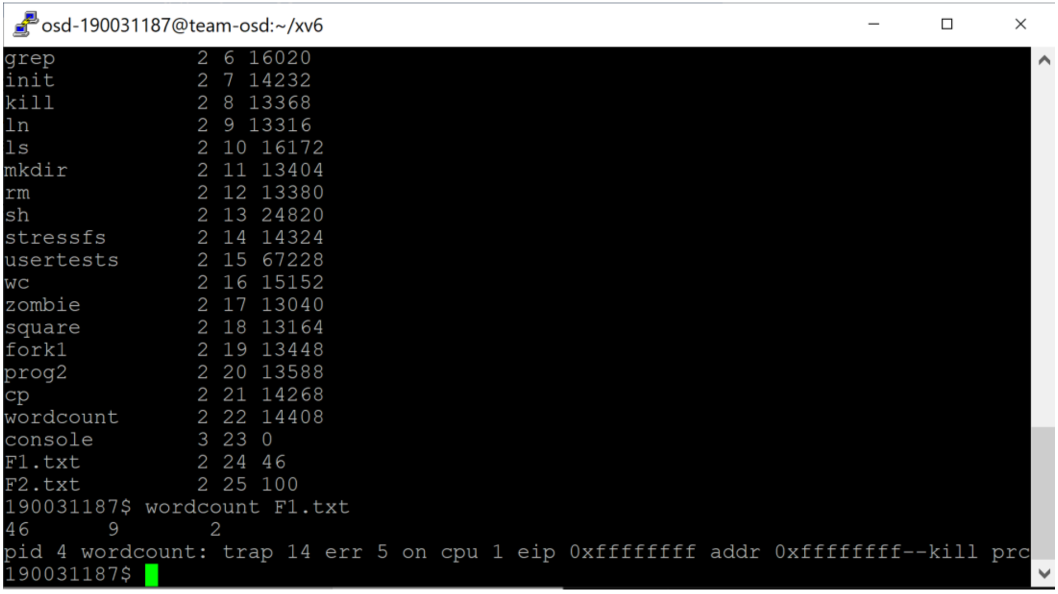
9. Type make qemu-nox

10. check whether the commands are working or not.

OUTPUT



```
SeaBIOS (version 1.11.0-2.el7)


iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF94780+1FED4780 C980



Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap sta8
init: starting sh
190031187$ cat>F1.txt
my name is radhakrishna
my id no is 190031187
190031187$ cp F1.txt F2.txt
pid 4 cp: trap 14 err 5 on cpu 1 eip 0xffffffff addr 0xffffffff--kill proc
190031187$ cat F2.txt
my name is radhakrishna
my id no is 190031187
190031187$
```



```
grep           2 6 16020
init           2 7 14232
kill           2 8 13368
ln             2 9 13316
ls             2 10 16172
mkdir          2 11 13404
rm             2 12 13380
sh             2 13 24820
stressfs       2 14 14324
usertests      2 15 67228
wc             2 16 15152
zombie         2 17 13040
square         2 18 13164
fork1          2 19 13448
prog2          2 20 13588
cp             2 21 14268
wordcount      2 22 14408
console        3 23 0
F1.txt         2 24 46
F2.txt         2 25 100
190031187$ wordcount F1.txt
46      9       2
pid 4 wordcount: trap 14 err 5 on cpu 1 eip 0xffffffff addr 0xffffffff--kill prc
190031187$
```